

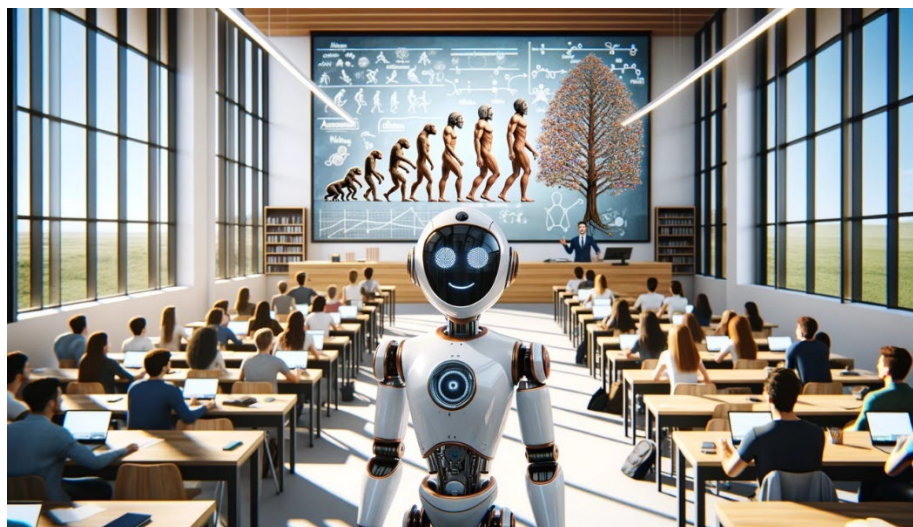


INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

AULA 2



Prof. Elton Masaharu Sato



Crédito: Elton Sato/Dall-E.

Imagine um mundo em que as soluções para problemas complexos emergem não através de um pensamento linear, mas evoluem naturalmente, assim como as espécies em nosso planeta. Essa não é uma premissa de um livro de ficção científica, mas uma realidade palpável no campo da computação, graças ao fascinante conceito dos algoritmos genéticos. Nesta abordagem, vamos embarcar em uma jornada intelectual que nos levará desde o vibrante e intrincado metrô de Tóquio até as profundezas do código que impulsiona essas inovações evolutivas.

Esta abordagem contemplará os seguintes tópicos:

- inspiração do algoritmo genético;
- criação de uma nova população;
- teste de *fitness*;
- *crossover* e mutação;
- *loop* iterativo e parâmetros.

TEMA 1 – INSPIRAÇÃO DO ALGORITMO GENÉTICO

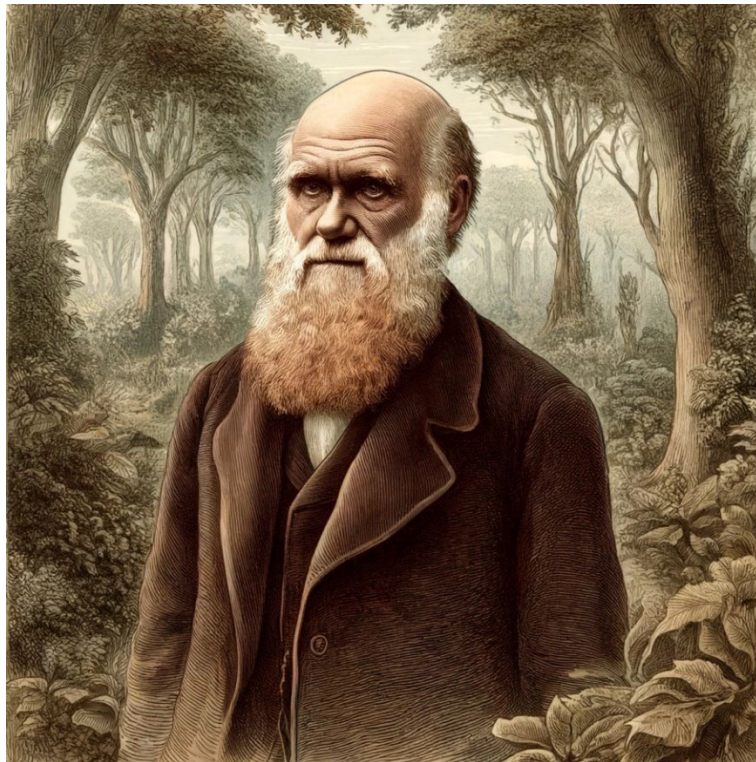
1.1 Origem e inspiração

Os algoritmos genéticos (AGs) são uma fascinante manifestação da tentativa humana de emular os processos da natureza para resolver problemas complexos em computação. Inspirados pela teoria da evolução de Charles Darwin, esses algoritmos são fundamentados no princípio de seleção natural,



segundo o qual os indivíduos mais aptos têm maiores chances de sobreviver e reproduzir. A ideia básica é simples, porém poderosa: assim como na natureza, onde as espécies evoluem ao longo de gerações para se adaptar a seus ambientes, os AGs desenvolvem soluções para problemas através de processos iterativos de seleção, cruzamento (crossover) e mutação.

Figura 1 – Retrato de Charles Darwin em uma floresta



Crédito: Elton Sato/Dall-E.

Apesar de existir uma proposta descrita por Alan Turing, em 1950, para uma máquina que aprenderia de forma paralela aos princípios da evolução, e de alguns outros pesquisadores terem criado modelos e máquinas que tentariam simulá-los, o desenvolvimento dos primeiros conceitos formais que deram origem aos algoritmos genéticos como os conhecemos é frequentemente creditado a John Henry Holland, na Universidade de Michigan, durante a década de 1960. Ele buscava não apenas resolver problemas específicos, mas também entender os processos adaptativos complexos que governam a evolução natural. Holland popularizou a área com o livro *Adaptation in Natural and Artificial Systems* (adaptação em sistemas naturais e artificiais, em português).

Para aqueles que não se recordam, a evolução das espécies depende de três fatores fundamentais, descritos a seguir.



- **Variação genética:** na natureza, cada indivíduo de uma espécie tem um conjunto único de características. Essa diversidade surge através de mutações (alterações aleatórias no DNA) e *crossover* (o cruzamento de características genéticas entre dois pais durante a reprodução). Exemplo: para plantas, como o milho, é muito importante que haja uma variedade genética, pois, caso uma praga venha a atingir essa espécie, alguns indivíduos serão capazes de resistir mais a ela, passando para a frente essa resistência.

1.2 Exemplos e aplicações

Toyota Prius: o sistema de transmissão deste modelo foi melhorado com a utilização de um algoritmo genético, otimizando sua eficiência e reduzindo em 30% o consumo de combustível deste carro híbrido (Biros; Kyslan; Durovsky, 2017).

Figura 2 – Um Toyota Prius aberto para demonstração com o sistema de transmissão visível



Crédito: Karolis Kavolelis/Shutterstock.

Boeing 787: utilizando algoritmos genéticos, os pesquisadores conseguiram criar um modelo para previsão de consumo de combustível em relação à altitude do veículo durante a descida (Turgut; Rosen, 2012).



Figura 3 – O Boeing modelo 787 Dreamliner revolucionou o custo de voos internacionais



Crédito: Minh K Tran/Shutterstock.

Antena da nave espacial ST5 da Nasa: com o avanço dos algoritmos genéticos, é possível criar antenas com formatos complexos e assimétricos que têm *performance* superior a qualquer outra antena desenvolvida manualmente (Hornby et al., 2006).

Figura 4 – Uma antena para bandas X da nave espacial da Nasa



Fonte: Wikimedia Commons/Nasa.



Fluoxetina (Prozac): durante o desenvolvimento do antidepressivo também foram utilizados algoritmos genéticos para determinar qualidades e melhora da *performance*. O medicamento também é usado para tratar depressão em animais, como cães, e animais em zoológicos, tudo com a supervisão de um veterinário (Nezhadali; Motlagh; Sadeghzadeh, 2018).

Figura 5 – Medicamento antidepressivo Prozac

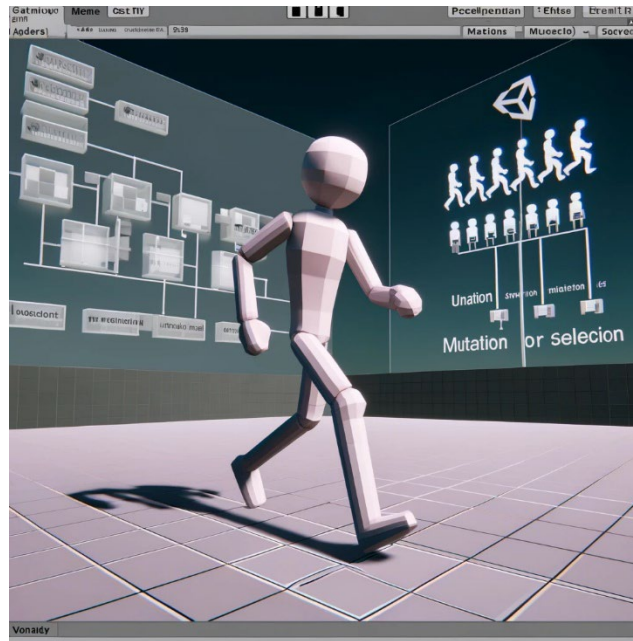


Crédito: Simone Hogan/Shutterstock.

Videogames: em *videogames* o algoritmo genético tem muitos usos, desde a geração de animação e movimento de modelos 3D até a descoberta de novas estratégias por jogadores, como o 7 Roach Rush, uma estratégia descoberta por algoritmo genético para o jogo StarCraft 2, divulgada em um fórum do time competitivo TeamLiquid. A estratégia consistia em criar essa tropa em uma quantidade de tempo que não se considerava possível até sua descoberta pela IA. Por um tempo, foi considerada uma estratégia imbatível, até em conversas e discussões em canais oficiais do jogo, questionando-se se existia alguma possibilidade de um jogador sobreviver a isso (Goatstream, 2013; Smith, 2010).



Figura 6 – Modelo 3D aprendendo a andar na *engine* de jogos Unity



Crédito: Elton Sato/Dall-E.

Pontos importantes

- Qual é a inspiração dos algoritmos genéticos?
- Algoritmos genéticos abrangem diversas áreas.
- Relação entre inteligência artificial e darwinismo.
- O que é variação genética, seleção natural e hereditariedade e quais são suas vantagens?

TEMA 2 – CRIAÇÃO DE UMA NOVA POPULAÇÃO



Crédito: Elton Sato/Dall-E.



No universo dos algoritmos genéticos, a criação de uma nova população é um processo-chave que dá início à jornada evolutiva em busca da solução ótima para um dado problema. A primeira etapa é a criação da população inicial, e ela é crucial, pois estabelece a diversidade genética que permitirá ao algoritmo explorar eficazmente o espaço de soluções.

2.1 Conceitos fundamentais

Gene: um gene é a menor unidade de informação de um algoritmo genético e seu conjunto forma um cromossomo. Em um algoritmo genético, ele representa um aspecto específico da solução, como um parâmetro ou uma variável. Exemplo: mantendo nossa analogia com animais, pense que um gene seria responsável por uma única característica de um animal, como ter ou não ter garras, a cor do olho, ou poder ou não voar.

Neste momento, você pode se perguntar: “Mas poder ou não voar não dependeria de outras características, como ter asas?”. É uma ótima pergunta. Como estamos falando de simulações de computador, não precisamos nos prender às leis da física da realidade, mas o quão real ou não é a simulação depende do que o programador precisa. Lembremos que é preciso fazer esse balanço, pois quanto mais realista, mais complexo e lento fica o processo. Porém, ao simplificar demais, pode ser que não se consiga o resultado desejado!

Cromossomo: um cromossomo é uma representação de uma solução possível para o problema em questão. É composto por um vetor de genes, estruturados de forma a representar todas as variáveis e suas combinações possíveis que definem uma solução. Exemplo: digamos que nós queremos um cromossomo que defina a cor de um animal. Nós precisaríamos de, pelo menos, três genes: 1 para vermelho, 1 para azul e 1 para amarelo. Caso queiramos um animal com a pele verde, devemos ter um cromossomo $(0, 1, 1)$, em que haja 0 de vermelho, 1 de azul e 1 de amarelo.

Note que esse cromossomo de cores é a solução para nosso problema de cores! Portanto, cada cromossomo possível, é uma solução diferente, alguns melhores e outros piores. **População:** a população é um conjunto de indivíduos, cada um com seu cromossomo. Inicialmente, essa população é gerada aleatoriamente ou por meio de algum critério específico, representando um amplo espectro de soluções possíveis. O tamanho da população pode variar,



dependendo do problema e do poder computacional disponível, mas deve ser grande o suficiente para garantir a diversidade genética. Exemplo: utilizando novamente o exemplo do milho, uma população de plantas representa um conjunto de muitos pés de milho, e cada um terá o próprio cromossomo, com similaridades e diferenças em relação a outros pés de milho.

Para este exemplo, imaginemos que para um milho ser mais resistente a um tipo específico de praga, ele deverá produzir menos espigas (ou seja, quanto mais resistente um milho for, menos ele irá se multiplicar). Cada indivíduo teria resistências diferentes para tipos de pragas específicas, e em um ambiente cheio de pragas apenas os milhos mais resistentes sobreviveriam, enquanto em um ambiente quase sem pragas ser muito resistente só reduz sua multiplicação.

Para uma população, é muito importante que haja uma variedade grande o suficiente para que os tipos certos de Milho possam se multiplicar.

2.2 Os passos da geração de uma população

1. **Definição do tamanho da população:** o primeiro passo na criação de uma nova população é decidir o tamanho da população. Esse é um parâmetro crítico, pois uma população muito pequena pode não oferecer diversidade genética suficiente, enquanto uma população muito grande pode tornar o processo evolutivo desnecessariamente lento e custoso computacionalmente.

2. **Geração de indivíduos:** cada indivíduo da população é gerado aleatoriamente ou através de heurísticas específicas. Esses indivíduos representam soluções potenciais para o problema em questão e são codificados de maneira apropriada, geralmente através de *strings* de *bits*, números reais ou qualquer outro método que melhor adeque-se ao problema.

3.

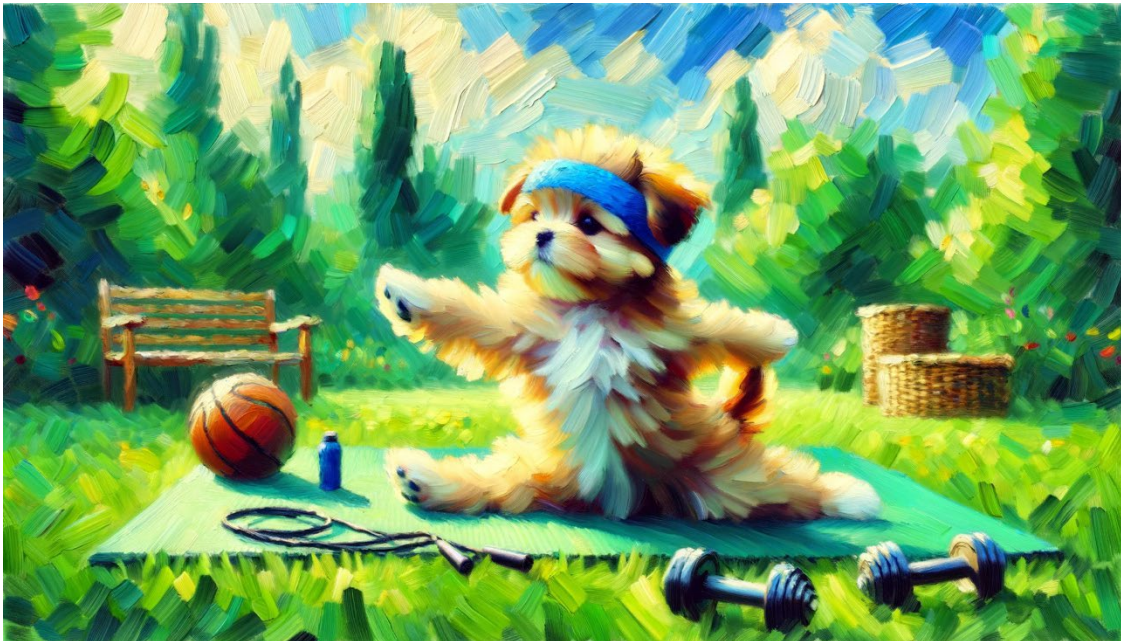
4.

Pontos importantes

- Definição de gene, cromossomo e população.
- Etapas da geração da população.

- Importância de uma variedade genética.

TEMA 3 – TESTE DE FITNESS



Crédito: Elton Sato/Dall-E.

Até o momento, vimos como gerar uma população e entendemos como algumas características da população são importantes, como sua variedade genética. Mas agora vem a etapa da seleção natural. Como faremos uma seleção natural de nossos indivíduos? Como saberemos qual dos indivíduos é o mais adaptado para o nosso problema e irá sobreviver e se reproduzir? É aí que entra o teste de fitness.

Fitness, do inglês, significa “aptidão” ou “adequação”. Dentro da área de algoritmos genéticos também usamos o termo performance ou desempenho como equivalente de fitness. A ideia é que quanto maior o fitness de um indivíduo, mais adaptado e adequado ele está para um ambiente. E o nosso objetivo é tentar descobrir o indivíduo com o maior fitness possível.

Se formos bastante técnicos, o fitness é uma métrica quantitativa que reflete o desempenho de cada indivíduo dentro da população, guiando a seleção natural do algoritmo na seleção dos candidatos mais promissores para reprodução. Ao determinar o quão fit ou apto um indivíduo é, o algoritmo pode priorizar aqueles que estão mais próximos de uma solução ótima, facilitando a evolução da população em direção a melhores respostas ao problema em questão.



Crédito: Elton Sato/Dall-E.

3.1 Como definir e realizar os testes de *fitness*

De forma simplificada, o teste de *fitness* é o problema que você está tentando resolver, então quanto maior o *fitness* de um indivíduo, mais próximo ele está de resolver o seu problema.

Para definirmos o teste de *fitness* na programação, devemos criar uma forma de quantificar o problema que queremos resolver. Normalmente o teste é programado em uma função que calcula o valor de cada indivíduo, com o objetivo de descobrir os indivíduos com o melhor *fitness*.



Crédito: Elton Sato/ Dall-E.



Exemplo: digamos que você está gerenciando uma barraca de frutas e tem dez moedas para comprar as frutas que quer revender em qualquer combinação das seguintes:

- maçã: custa 1 moeda e vende-se por 2 moedas;
- banana: custa 3 moedas e vende-se por 7 moedas;
- mamão: custa 8 moedas e vende-se por 20 moedas.

Considerando que a função de *fitness* é a seguinte: (moedas iniciais – moedas gastas + moedas recebidas), com a condição de que o total de moedas gastas não pode exceder dez moedas, qual o valor de *fitness* (em moedas) de cada um dos seguintes indivíduos:

- a. 2 maçãs, 2 bananas e 0 mamão?
- b. 0 maçã, 0 banana e 0 mamão?
- c. 0 maçã, 1 banana e 1 mamão?

Para **a**, nós temos 2 maçãs, que custam 2 e vendem-se por 4; 2 bananas, que custam 6 e vendem-se por 14; e 0 mamão, que custa 0 e vende-se por 0. Logo, temos (10 moedas iniciais – 8 moedas gastas + 18 moedas recebidas), um total de **20**. E as moedas gastas não excederam 10 moedas. Portanto, podemos dizer que o *fitness* de **a** é **20**.

Para **b**, nós temos 0 maçã, que custa 0 e vende-se por 0; 0 banana, que custa 0 e vende-se por 0; e 0 mamão, que custa 0 e vende-se por 0. Logo, temos (10 moedas iniciais – 0 moeda gasta + 0 moeda recebida) um total de **10**. E as moedas gastas não excederam 10 moedas. Portanto, podemos fazer que o *fitness* de **b** é **10**.

Para **c**, nós temos 0 maçã, que custa 0 e vende-se por 0; 1 banana, que custa 3 e vende-se por 7; e 1 mamão, que custa 8 e vende-se por 20. Logo, temos (10 moedas iniciais – 11 moedas gastas + 27 moedas recebidas), um total de **26**. Porém, as moedas gastas excederam 10 e, nesse caso, essa não é uma solução válida para o nosso problema e devemos descartá-la. O valor de *fitness* dessa solução, então, deve ser um valor bem baixo, para reduzirmos as chances de ele ser escolhido pela seleção natural. **0 (zero)** é um valor de *fitness* apropriado para punirmos os indivíduos que não apresentam soluções válidas, mas é possível também darmos um valor ainda menor, como **-999**, para garantir que, mesmo que tenhamos uma solução de valor 0 válida, a solução válida ainda tenha um *fitness* maior que a inválida.



3.2 Escolhendo indivíduos com base no *fitness*

A escolha de indivíduos com melhores *fitness* em um algoritmo genético é fundamental para direcionar a busca por soluções ótimas no espaço de soluções do problema. Após o cálculo do *fitness*, que quantifica a adequação de cada indivíduo (solução) ao problema, utiliza-se um método de seleção para determinar quais indivíduos serão pais da próxima geração.

Existem vários métodos de seleção, e a escolha depende do problema específico e das características desejadas na população evoluída. Listamos a seguir alguns dos métodos mais comuns.

- **Seleção por roleta:** este método é bem simples; usam-se os valores de *fitness* de cada solução para colocar todos os indivíduos em uma roleta.

Exemplo:

- a. indivíduo A tem 100 *fitness*;
- b. indivíduo B tem 60 *fitness*;
- c. indivíduo C tem 30 *fitness*;
- d. indivíduo D tem 10 *fitness*.

Se considerarmos que esses quatro são todos os indivíduos de uma população, ao criarmos uma roleta em que cada um tem o próprio *fitness* como tamanho da sua fatia da roleta, teríamos o seguinte: ($100 + 60 + 30 + 10 = 200$). Ou seja, o indivíduo A teria 50% de chances de ser escolhido para reproduzir, B teria 30% de chances de ser escolhido, C teria 15% e D, 5%.

- **Seleção por *ranking*:** este método é bastante similar ao da roleta, mas, antes de se criar a roleta, faz-se um *ranking* dos indivíduos e são atribuídos pontos fixos para cada colocação, utilizando-se então essa pontuação para fazer a roleta. Exemplo:

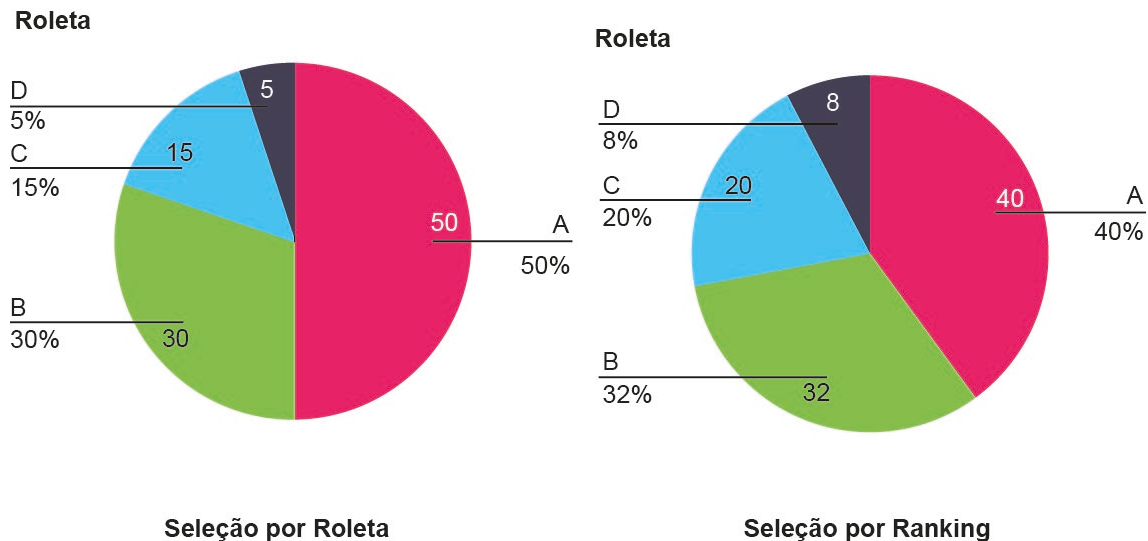
- a. indivíduo A tem 100 *fitness*;
- b. indivíduo B tem 60 *fitness*;
- c. indivíduo C tem 30 *fitness*;
- d. indivíduo D tem 10 *fitness*.

Digamos que o melhor *fitness* ganhasse 10 pontos, o segundo melhor ganhasse 8 pontos, o terceiro ganhasse 5 pontos e o quarto, 2 pontos.



Se considerarmos que esses quatro são todos os indivíduos de uma população, ao criarmos uma roleta em que cada um tem a sua pontuação como tamanho da sua fatia da roleta, teríamos o seguinte: $(10 + 8 + 5 + 2 = 25)$. Ou seja, o indivíduo A teria 40% de chances de ser escolhido para reproduzir, B teria 32% de chances de ser escolhido, C teria 20% e D, 8%.

Figura 7 – Roletas das seleções



Fonte: Elton Sato.

- **Elitismo:** além das escolhas aleatórias, elitismo é uma opção possível a ser usada em conjunto, que é permitir que o(s) melhor(es) indivíduo(s) passe(m) intacto(s) para a próxima geração, garantindo que em nenhum momento o melhor *fitness* da geração se reduza. Porém, é importante sempre usarmos isso em conjunto com alguns dos outros tipos de seleção, pois auxilia o algoritmo a não ficar preso em uma solução boa, mas que não seja a melhor possível. No exemplo das frutas, o algoritmo pode ficar preso com indivíduos que só compram maçãs.

Pontos importantes

- Definição de gene, cromossomo e população.
- Etapas da geração da população.
- Importância de uma variedade genética.



Cr dito: Elton Sato/Dall-E.

No cora o dos AGs est o os conceitos de *crossover* e muta o. Esses mecanismos s o inspirados nos processos biol gicos de reprodu o e varia o gen tica, que permitem a diversidade e a adapta o das esp cies ao longo das gera es. Essencialmente, o *crossover* e a muta o s o as for as motrizes que possibilitam que uma popula o de solu es explore o espa o de busca de maneira eficaz, evoluindo em dire o a solu es  timas ou satisfat rias.

Como vimos no in cio desta abordagem, o *crossover*   o cruzamento de caracter sticas gen ticas de dois indiv duos, enquanto a muta o   a altera o aleat ria dos genes.

4.1 Crossover

O *crossover*, ou cruzamento,   o processo pelo qual dois indiv duos (pais) combinam partes de seus c digos gen ticos para produzir um ou mais descendentes, com caracter sticas gen ticas derivadas de ambos. Esse mecanismo   fundamental para introduzir novas combina es gen ticas na popula o, promovendo a variabilidade gen tica necess ria para a evolu o das solu es.



De forma simplificada, é o motivo pelo qual você tem características que o tornam uma combinação das características de seu pai com as características de sua mãe.

Existem várias técnicas diferentes de realização do *crossover* em um código. Listamos aqui as mais comuns.

- **Crossover de um ponto:** para cada par de indivíduos (que chamaremos de A e B), é escolhido um único ponto. A partir disso, criam-se dois indivíduos, um que possui todos os genes de A até o ponto escolhido e depois todos os genes de B, e outro indivíduo que é o contrário: ele possui todos os genes de B até o ponto escolhido e depois todos os genes de A. Exemplo: digamos que o indivíduo A tem o cromossomo **aaaaa** e o indivíduo B tem o cromossomo **bbbbbb**. Com cinco genes, nós temos seis lugares diferentes para escolher como ponto:

1 a 2 a 3 a 4 a 5 a 6

Caso escolhamos o ponto 2, nossos dois novos indivíduos serão todos os genes de A até o ponto 2 e, depois, todos os genes de B:

aabbb

e todos os genes de B até o ponto 2 e, depois, todos os genes de A:

bbaaa..

- **Crossover de dois pontos:** funciona de forma similar ao *crossover* de um ponto, mas o processo é feito mais uma vez para um segundo ponto. Ou seja, criam-se dois indivíduos, um que terá todos os genes de A até o primeiro ponto, depois todos os genes de B até o segundo ponto e então volta para os genes de A. O outro indivíduo será o contrário deste. Exemplo: digamos que o indivíduo A tem o cromossomo **aaaaa** e o indivíduo B tem o cromossomo **bbbbbb**. Com cinco genes, nós temos seis lugares diferentes para escolher como ponto:

1 a 2 a 3 a 4 a 5 a 6

Agora escolhemos dois pontos, por exemplo, o ponto 2 e o 5, todos os genes de A até o ponto 2, depois todos os genes de B até o ponto 5 e, então, todos os genes de A:

aabba



e todos os genes de B até o ponto 2, depois todos os genes de A até o ponto 5 e, então, todos os genes de B:

bbaab

- **Crossover uniforme:** este é o mais simples de todos os *crossovers*. Para cada gene do cromossomo, escolhemos aleatoriamente se o novo indivíduo criado receberá o gene de A ou de B e, no final, criamos um segundo indivíduo que é o contrário dele.

Exemplo: digamos que o indivíduo A tem o cromossomo **aaaaa** e o indivíduo B tem o cromossomo **bbbbbb**. Para o primeiro indivíduo, escolhemos aleatoriamente genes de A ou de B:

baaba

Depois, criamos um segundo indivíduo que é o contrário dele:

abbab.

4.2 Mutação

O *crossover* faz um trabalho excelente ao misturar informações genéticas existentes para criar indivíduos com altas probabilidades de ter *fitness* similar ao de seus pais.

Porém, o *crossover* sozinho tem uma fraqueza: ele não inclui informações **novas** na população; a mutação introduz novas informações ao modificar aleatoriamente os genes dos indivíduos. Embora ocorra com menor frequência que o *crossover*, a mutação é crucial para manter a diversidade genética dentro da população e para evitar a convergência prematura para soluções subótimas.

Apesar de existirem diversas técnicas diferentes para realização da mutação, a mais comum é a mutação uniforme, também chamada de *mutação pontual*, em que, de gene em gene, decidimos aleatoriamente se iremos fazer a mutação daquele gene ou não.

Exemplo: digamos que temos o seguinte indivíduo:

abbabbaabbaabbbbaaabaa

E fazemos uma mutação com um valor extremamente alto, como 10%, por exemplo. Podemos ter o seguinte indivíduo:

ybbabbaabbaabbbbaaabab

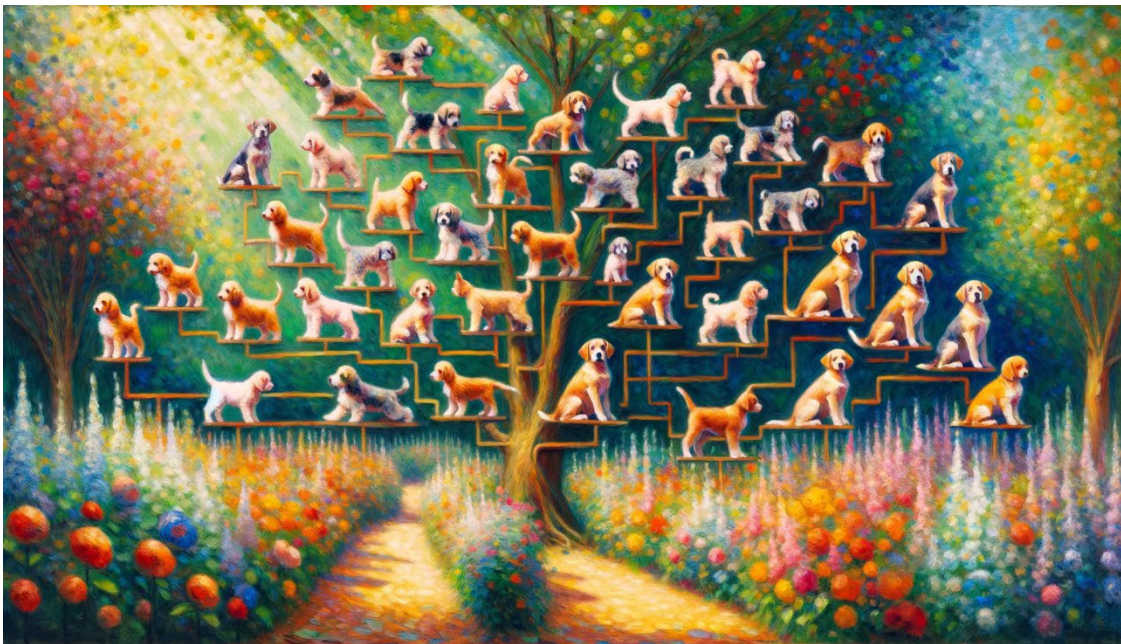


A cada gene desse indivíduo, eu decidi, aleatoriamente, com 10% de chances, se eu iria alterar aquele gene para um valor aleatório. Embora o cromossomo tenha se mantido praticamente igual ao de antes, a mutação introduziu informações novas que podem levar a indivíduos com melhor *fitness*.

Pontos importantes

- O que é *crossover*?
- Quais são os tipos de *crossover*?
- O que é mutação?
- Para que serve o *crossover*?
- Para que serve a mutação?

TEMA 5 – LOOP ITERATIVO E PARÂMETROS



Crédito: Elton Sato/Dall-E.

5.1 Loop iterativo

Até este ponto de nossa jornada pelo universo dos algoritmos genéticos, exploramos diversos conceitos fundamentais – desde a mecânica de criação de novas populações, passando pelo teste de *fitness* para determinar os melhores indivíduos, até os processos de *crossover* e mutação, criando indivíduos que possuem soluções baseadas nas anteriores e com um toque de novidade.



Porém, falta falarmos de uma parte fundamental do algoritmo genético, que é seu caráter iterativo, ou seja, ele faz iterações, repetindo as etapas de *fitness*, *crossover* e mutação até atingirmos um critério de parada. Cada etapa é chamada de *uma geração*, e a execução de um algoritmo genético pode ter quantas gerações forem necessárias – dezenas, centenas ou milhares.

O critério de parada varia de problema para problema, mas, em geral, um dos seguintes critérios é escolhido:

- **número fixo de gerações:** de forma simples, é decidida uma quantidade fixa de gerações, como, por exemplo, cem; o algoritmo é rodado até chegar a cem gerações, e então pega-se a solução com melhor *fitness* gerado;
- **valor de *fitness*-alvo:** assim como o próprio nome diz, o algoritmo é executado até um indivíduo atingir ou superar um *fitness* definido no começo; é importante notar que esse critério pode fazer com que o algoritmo rode para sempre;
- **convergência de *fitness*:** esse critério faz com que o algoritmo pare de executar assim que ele detectar que o *fitness* convergiu; isso normalmente significa que durante uma certa quantidade de gerações não houve nenhuma melhora no melhor *fitness* encontrado.

5.2 Parâmetros

Este é um tópico que será bastante comum entre todos os tipos de inteligência artificial: os parâmetros utilizados. Embora cada tipo de inteligência artificial possua os próprios parâmetros, é importante entendê-los bem, já que a eficiência, ou até mesmo o funcionamento da inteligência artificial, está baseada nos parâmetros utilizados para cada uma delas.

A melhor forma de aprender sobre como os parâmetros influenciam uma inteligência artificial é realizar testes empíricos e analisar seus resultados. Por isso recomendo a você que teste todas as inteligências artificiais aqui mencionadas e mexa nos parâmetros, analisando as diferenças entre as execuções. No caso dos algoritmos genéticos, pode ser uma diferença de tempo de execução, quantidade de gerações ou até mesmo qualidade do resultado atingido.



Alguns dos parâmetros a serem observados:

- tamanho da população inicial;
- função de *fitness*;
- método de seleção;
- se terá elitismo; se sim, quantos passarão para a frente;
- taxa de *crossover*;
- tipo de *crossover*;
- taxa de mutação;
- critério de parada.

Pontos importantes

- O *loop* iterativo dos algoritmos genéticos.
- Os critérios de parada dos algoritmos genéticos.
- Importância de analisar os parâmetros de uma inteligência artificial.

FINALIZANDO

Nesta abordagem, aprendemos sobre a mecânica e a aplicabilidade dos algoritmos genéticos e sobre a filosofia por trás dessa tecnologia: a ideia de que podemos nos inspirar nos processos naturais para enfrentar e superar desafios computacionais e técnicos. Essa abordagem reforça a concepção de que a tecnologia, especialmente no campo da inteligência artificial, não precisa ser vista como uma entidade separada do mundo natural ou oposta a ele, mas como uma extensão da nossa compreensão de processos naturais adaptada para servir a propósitos humanos.



REFERÊNCIAS

BIROS, M.; KYSLAN, K.; DUROVSKY, F. Optimization of hybrid vehicle drivetrain with genetic algorithm using Matlab and advisor. **Journal of Engineering Science and Technology Review**, v. 10, p. 35-40, 2017.

GOATSTREAM. **Flexible Muscle-Based Locomotion for Bipedal Creatures**. 25 nov. 2013. Disponível em: <<https://youtu.be/pgxEE27nsQw>>. Acesso em: 16 abr. 2024.

HORNBY, G. et al. Automated Antenna Design with Evolutionary Algorithms. **American Institute of Aeronautics and Astronautics – Space**, p. 1-8, 2006.

NEZHADALI, A.; MOTLAGH, M. O.; SADEGHZADEH, S. Spectrophotometric determination of fluoxetine by molecularly imprinted polypyrrole and optimization by experimental design, artificial neural network, and genetic algorithm. **Spectrochim Acta A Mol Biomol Spectrosc**, n. 190, p. 181-187, 2018.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: a Modern Approach**. Londres: Pearson, 2010.

SMITH, Q. Algorithms Discover Build Order From Hell. **Rock Paper Shotgun**. 2 nov. 2010. Disponível em: <<https://www.rockpapershotgun.com/genetic-algorithms-find-build-order-from-hell>>. Acesso em: 16 abr. 2024.

TURGUT, E. T.; ROSEN, M. A. Relationship between fuel consumption and altitude for commercial aircraft during descent: preliminary assessment with a genetic algorithm. **Aerosp. Sci. Technol**, v. 17, n. 1, p. 65-73, 2012.

WOOLDRIDGE, M. **An Introduction to Multiagent Systems**. Chichester: John Wiley & Sons, 2009.

_____. **A Brief History of Artificial Intelligence: What It Is, Where We Are, and Where We Are Going**. Chichester: John Flatiron Books, 2021.